An Analysis of First-Order Logics of Probability

Joseph Y. Halpern
IBM Almaden Research Center, Dept. K53/802
650 Harry Rd.
San Jose, CA 95120
email: halpern@ibrn.com

Abstract

We consider two approaches to giving semantics to first order logics of probability. The first approach puts a probability on the domain, and is appropriate for giving semantics to formulas involving statistical information such as "The probability that a (typical) bird flies is greater than .9." The second approach puts a probability on possible worlds, and is appropriate for giving semantics to formulas describing degrees of belief, such as "The probability that Tweety (a particular bird) flies is greater than .9." We show that the two approaches can be easily combined, allowing us to reason in a straightforward way about statistical information and degrees of belief. We then consider axiornatizing these logics. In general, it can be shown that no complete axiomatization is possible. We provide axiom systems that are sound and complete in cases where a complete axiomatization is possible, showing that they do allow us capture a great deal of interesting reasoning about probability.

1 Introduction

Consider the two statements "I he probability that a bird chosen at random will fly is greater than .9" and "The probability that Tweety (a particular bird) flies is greater than .9." It is quite straightforward to capture the sec ond statement by using a possible-world semantics along the lines of that used in [FH88, FHM88, Nil8f>]. Namely, we can imagine a number of possible worlds such that the predicate *Flies* has a different extension in each one. Thus, *Flies(Tweety)* would hold in some possible worlds, and not in others. We then put a probability distribution on this set of possible worlds, and check if the set of possible worlds where *Flics(Tweety)* holds has probability greater than .9.

However, as pointed out by Bacchus [Bac88b, Bac88a], this particular possible worlds approach runs into difficulties when trying to represent the first statement, which we may believe as a result of statistical information of the form "More than 90% of all birds fly." What is the formula that should hold at a set of worlds whose probability is greater than .9? The most obvious candidate is perhaps *Vx(Btrd(x) => Flies(x)).* However, it might very well be the case that in each of the worlds we consider possible, there is at least one bird that doesn't fly. Hence, the statement *Wx(Bird(x) => Fltes(x))* holds in none of the worlds (and so has probability 0). Thus it cannot be used to represent the statistical information. As Bacchus shows, other straightforward approaches do not work either.

There seems to be a fundamental difference between these two statements. The first captures statistical information, and the second captures what has been called a *degree of belief* [Bac88b, Kyb88]. The first statement seems to assume only one possible world (the "real" world), and in this world, some probability distribution over the set of birds. It is saying that if we consider a bird chosen at random, then with probability greater than .9 it will fly. The second statement implicitly assumes the existence of a number of possibilities (in some of which Tweety flies, while in others Tweety doesn't), with some probability over these possibilities.

Bacchus [Bac88b] provides a syntax and semantics for a first order logic for reasoning about statistical information, where the probability is placed on the domain. This approach has difficulties dealing with degrees of belief. For example, if there is only one fixed world, in this world either Tweety flies or he doesn't, so Flics(Tweely)*)* holds with either probability 1 or probability 0. In particular, a statement such as "The probability that Tweety flies is between .9 and .95" is guaranteed to be false! Recognizing this difficulty, Bacchus moves beyond the syntax of his logic to define the notion of a *belief Junction,* which lets us talk about the degree of belief in the formula *a* given a knowledge base *ft.* However, it would clearly be useful to be able to capture reasoning about degrees of belief within a logic, rather than moving outside the logic to do so.

Tn this paper, we describe two first-order logics, one for capturing reasoning about statistical information, and another for reasoning about degrees of belief. We then show how the two can be easily combined in one frame work, allowing us to simultaneously reason about statistical information and degrees of belief.

We go on to consider issues of axiomatizability. Bacchus is able to provide a complete axiomatization for his language because he allows probabilities to take on *non-standard* values in arbitrary ordered fields. Results of a companion paper [AH89] show that if we require probabilities to be real-valued (as we do here), we cannot in general hope to have a complete axiomatization for our language. We give sound axiom systems here which we show are complete for certain restricted settings. This suggests that our axiom systems are sufficiently rich to capture a great deal of interesting reasoning about probability.

2 Probabilities on the domain

We assume that we have a first-order language for reasoning about some domain. We take this language to consist of a collection Φ of predicate symbols and function symbols of various arities (as usual, we can identify constant symbols with functions symbols of arity 0). Given a formula φ in the logic, we also allow formulas of the form $w_x(\varphi) \geq 1/2$, which can be interpreted as "the probability that a random x in the domain satisfies φ is greater than or equal to 1/2". We actually extend this to allow arbitrary sequences of distinct variables in the subscript. To understand the intuition behind this, suppose the formula $Son(x,y)$ says that x is the son of y. Now consider the three terms $w_x(Son(x,y))$, $w_y(Son(x,y))$, and $w_{(x,y)}(Son(x,y))$. The first describes the probability that a random x is the son of y; the second describes the probability that x is the son of a random y; the third describes the probability that a random pair (x,y) will have the property that x is the son of y.

We formalize these ideas by using a two-sorted language. The first sort consists of the function symbols and predicate symbols in Φ, together with a countable family of *object variables* $x^o, y^o, \ldots$. The second sort consists of the binary function symbols $+$ and $\times$, which represent addition and multiplication, constant symbols 0 and 1, representing the real numbers 0 and 1, binary relation symbols $>$ and $=$, and a countable family of *field variables* $x^f, y^f, \ldots$. (We drop the superscripts on the variables when it is clear from context what sort they are.) Terms of the first sort describe elements of the domain we want to reason about. Terms of the second sort typically represent probabilities, which we want to be able to add and multiply.

We now define object terms, field terms, and formulas simultaneously by induction. We form object terms, which range over the domain of the first-order language, by starting with object variables and closing off under function application, so that if f is an n-ary function symbol in Φ and $t_1, \ldots, t_n$ are object terms, then $f(t_1, \ldots, t_n)$ is an object term. We form field terms, which range over the reals, by starting with 0, 1, and probability terms of the form $w_{\bar{x}}(\varphi)$, where φ is an arbitrary formula, and then closing off under $+$ and $\times$, so that $t_1 + t_2$ and $t_1 \times t_2$ are field terms if t_1 and t_2 are. We form formulas in the standard way. We start with *atomic formulas*: if P is an n-ary predicate symbol in Φ, and $t_1, \ldots, t_n$ are object terms, then $P(t_1, \ldots, t_n)$ is an atomic formula, while if t_1 and t_2 are field terms, then $t_1 = t_2$ and $t_1 > t_2$ are atomic formulas. We sometimes also consider the situation where there is an equality symbol for object terms; in this case, if t_1 and t_2 are object terms, then $t_1 = t_2$ is also an atomic formula. We then close off under conjunction, negation, and universal quantification, so that if φ_1 and φ_2 are formulas and x is a (field or object) variable, then $\varphi_1 \wedge \varphi_2$, $\neg\varphi_1$, and $\forall x \varphi_1$ are all formulas. We call the resulting language $\mathcal{L}_1(\Phi)$; if it includes equality between object terms, we call it $\mathcal{L}_1^=(\Phi)$.

We define $\vee$, $\Rightarrow$, and $\exists$, in terms of $\wedge$, $\neg$, and $\forall$ as usual. In addition, if t_1 and t_2 are two field terms, we use other standard abbreviations such as $t_1 < t_2$ for $t_2 > t_1$, $t_1 \geq t_2$ for $t_1 > t_2 \vee t_1 = t_2$, $t_1 \geq 1/2$ for $(1+1) \times t_1 \geq 1$, and so on.

The only differences between our syntax and that of Bacchus is that we write $w_{\bar{x}}(\varphi)$ rather than $[\varphi]_{\bar{x}}$, and, for simplicity, we do not consider what Bacchus calls *measuring functions* (functions which map object terms into field terms), and the only field functions we allow are $+$ and $\times$. The language is still quite rich, allowing us to express conditional probabilities, notions of independence, and statistical notions; we refer the reader to [Bac88b] for examples.

We define a *type 1 probability structure* to be a tuple (D, π, μ), where D is a domain, π assigns to the predicate and function symbols in Φ predicates and functions of the right arity over D (so that (D, π) is just a standard first-order structure), and μ is a discrete probability function on D. That is, we take μ to be a mapping from D to the real interval $[0,1]$ such that $\sum_{d \in D} \mu(d) = 1$. For any $A \subseteq D$, we define $\mu(A) = \sum_{d \in A} \mu(d)$.[1] Given a probability function μ, we can then define a discrete probability function μ^n on the product domain D^n consisting of all n-tuples of elements of D by taking $\mu^n(d_1, \ldots, d_n) = \mu(d_1) \times \ldots \times \mu(d_n)$. Define a *valuation* to be a function mapping each object variable into an element of D and each field variable into an element of $\mathbb{R}$ (the reals). Given a type 1 probability structure M and valuation v, we proceed by induction to associate with every object (resp. field) term t an element $[t]_{(M,v)}$ of D (resp. $\mathbb{R}$), and with every formula φ a truth value, writing $(M,v) \models \varphi$ if the value **true** is associated with φ by (M,v). The definitions follow the lines of first-order logic, so we just give a few clauses of the definition here, leaving the remainder to the reader:

[1] The restriction to discrete probability functions is made here for ease of exposition only. We can allow arbitrary probability functions on the domain by considering *inner measures*, as is done in [FHM88, FH89]. It might seem that for practical applications we should further restrict to *uniform* probability functions, i.e., ones that assign equal probability to all domain elements. Although we allow uniform probability functions, and the language is expressive enough to allow us to say that the probability on the domain is uniform, we do not require them. There are a number of reasons for this. For one thing, there are no uniform probability functions in countable domains. (Such a probability function would have to assign probability 0 to each individual element in the domain, which means by countable additivity it would have to assign probability 0 to the whole domain.) And even if we restrict attention to finite domains, we can construct two-stage processes (where, for example, one of three urns is chosen at random, and then some ball in the chosen urn is chosen at random) where the most natural way to assign probabilities would not assign equal probability to every ball [Car50].

- $(M, v) \models t_1 = t_2$ iff $[t_1]_{(M,v)} = [t_2]_{(M,v)}$.
- $(M, v) \models \forall x^o \varphi$ iff $(M, v[x^o/d]) \models \varphi$ for all $d \in D$, where $v[x^o/d]$ is the valuation which is identical to v except that it maps x^o to d.
- $[w_{\langle x_1,\ldots,x_n\rangle}(\varphi)]_{(M,v)} = \mu^n(\{(d_1,\ldots,d_n) : (M, v[x_1/d_1,\ldots,x_n/d_n]) \models \varphi\})$.

The major difference between our semantics and that of Bacchus is that Bacchus allows *nonstandard* probability functions, that take values in arbitrary ordered fields, and are only finitely additive, not necessarily countably additive. Our probability functions are standard: they are real-valued and countably additive. (Bacchus allows such nonstandard probability functions in order to obtain a complete axiomatization for his language. We return to this point later.)

We write $M \models \varphi$ if $(M, v) \models \varphi$ for all valuations v, and write $\models_1 \varphi$, and say that φ *is valid with respect to type 1 structures*, if $M \models \varphi$ for all type 1 probability structures M.

As an example, suppose the language has only one predicate, the binary predicate Son, and we have a structure $M = (\{a, b, c\}, \pi, \mu)$ such that $\pi(Son)$ consists of only the pair (a, b), $\mu(a) = 1/3$, $\mu(b) = 1/2$, and $\mu(c) = 1/6$. Let v be a valuation such that $v(x) = a$ and $v(y) = c$. Then it is easy to check that we have $[w_x(Son(x,y))]_{(M,v)} = 0$, $[w_y(Son(x,y))]_{(M,v)} = 1/2$, and $[w_{\langle x,y\rangle}(Son(x,y))]_{(M,v)} = 1/6$. Thus, if we pick a random x from the domain and fix y to be c, the probability that x is a son of y is 0: no member of the domain is a son of c. If we fix x to be a and pick a y at random from the domain, the probability that x is a son of y is $1/2$, which is exactly the probability that $y = b$. Finally, if we pick pairs at random from the domain, the probability of picking a pair (x, y) such that x is a son of y is $1/6$.

This example shows that the syntax and semantics of this logic are well suited for reasoning about statistical information. But, as discussed in the introduction, the logic is not well suited for making statements about degrees of belief about properties of particular individuals. It is easy to see that for any closed formula φ, such as $Fly(Tweety)$, we have $\models_1 w_{\vec{x}}(\varphi) = 0 \lor w_{\vec{x}}(\varphi) = 1$ for any vector $\vec{x}$ of distinct variables. Thus, a formula such as $.9 \leq w_{\vec{x}}(Flies(Tweety)) \leq .95$ is guaranteed to be false. In order to make sense out of such formulas, we must put probabilities on possible worlds.

3 Probabilities on possible worlds

We have seen that in a precise sense type 1 probability structures are inappropriate for reasoning about degrees of belief. In practice, it might well be the case that the way we derive our degrees of belief is from the statistical information at our disposal. For example, if we know that $w_x(Bird(x) \Rightarrow Flies(x)) \geq .9$ in a given structure, and we know that Tweety is a bird, then we might conclude that the probability that Tweety flies at least .9. However, as pointed out by Bacchus and others, this type of reasoning is fraught with difficulties. For example, if we have more specific information about Tweety, such as the fact that Tweety is a penguin, we no longer want to draw the conclusion that the probability that Tweety flies is at least .9. Bacchus provides some heuristics for deriving such degrees of belief. While this is a very interesting topic to pursue, it also seems appropriate to construct a formal model that allows us to directly capture such degrees of belief. Such a formal model can be constructed in a straightforward way using possible worlds, as we now show.

The syntax for a logic for reasoning about possible worlds is essentially the same as the syntax used in the previous section. Starting with a set Φ of function and predicate symbols, we form more complicated formulas and terms as before except that instead of allowing probability terms of the form $w_{\vec{x}}(\varphi)$, where $\vec{x}$ is some vector of distinct object variables, we only allow probability terms of the form $w(\varphi)$, interpreted as "the probability of φ". Since we are no longer going to put a probability distribution on the domain, it does not make sense to talk about the probability that a random choice for $\vec{x}$ will satisfy φ. It does make sense to talk about the probability of φ though: this will be the probability of the set of possible worlds where φ is true. We call the resulting language $\mathcal{L}_2(\Phi)$; if it includes equality between object terms, we call it $\mathcal{L}_2^=(\Phi)$.

More formally, a *type 2 probability structure* is a tuple (D, S, π, μ), where D is a domain, S is a set of *states* or *possible worlds*, for each state $s \in S$, $\pi(s)$ assigns to the predicate and function symbols in Φ predicates and functions of the right arity over D for each state $s \in S$, and μ is a discrete probability function on S. Note the key difference between type 1 and type 2 probability structures: in type 1 probability structures, the probability is taken over the domain D, while in type 2 probability structures, the probability is taken over the set S of states. Given a type 2 probability structure M, a state s, and valuation v, we can associate with every object (resp. field) term t an element $[t]_{(M,s,v)}$ of D (resp. $\mathbb{R}$), and with every formula φ a truth value, writing $(M, s, v) \models \varphi$ if the value **true** is associated with φ by (M, s, v). Note that we now need the state to provide meanings for the predicate and function symbols; they might have different meanings in each state. Again, we just give a few clauses of the definition here, which should suffice to indicate the similarities and differences between type 1 and type 2 probability structures:

- $(M, s, v) \models t_1 = t_2$ iff $[t_1]_{(M,s,v)} = [t_2]_{(M,s,v)}$.
- $(M, s, v) \models \forall x^o \varphi$ iff $(M, s, v[x^o/d]) \models \varphi$ for all $d \in D$,
- $[w(\varphi)]_{(M,s,v)} = \mu(\{s' \in S : (M, s', v) \models \varphi\})$.

We say $M \models \varphi$ if $(M, s, v) \models \varphi$ for all states s in M and all valuations v, and say φ is *valid with respect to type 2 structures*, and write $\models_2 \varphi$, if $M \models \varphi$ for all type 2 probability structures M.

As expected, in type 2 probability structures, it is completely consistent for the probability that Tweety flies to be between .9 and .95. A sentence such as $.9 \leq w(Flies(Tweety)) \leq .95$ is true in a structure M (independent of the state s) precisely if the set of states where $Flies(Tweety)$ is true has probability between .9

and .95. However, there is no straightforward way to capture statistical information using $\mathcal{L}_2$?

Possible extensions: We have made a number of simplifying assumptions in our presentation of type 2 probability structures. We now briefly discuss how they might be dropped.

1. Just as in the case of type 1 probability structures, we can allow arbitrary probability functions, not just discrete ones, by using inner measures.
2. We have assumed that all functions and predicates are *flexible*, i.e., they may take on different meanings at each state. We can easily designate some functions and predicates to be *rigid*, so that they take on the same meaning at all states.
3. We have assumed that there is only one domain. We could instead view each state s as a first-order structure, with its own domain D_s. In this case, $\pi(s)$ would assign to each predicate and function symbol in Φ a predicate (resp. function) of the right arity on D_s. Let $D = \cup_{s \in S} D_s \cup \{\perp\}$, where $\perp$ is a distinguished element not in $\cup_{s \in S} D_s$. We now take a valuation to be a function that associates with each field variable an element of $I\!R$, as before, and with each object variable an element of D. Given a predicate on D_s, we can view it as a predicate on D with the same extension. We can also extend an n-ary function f on D_s to one on D by defining $f(d_1, \ldots, d_n) = \perp$ if some $d_i \notin D_s$. We can now define the semantics of terms and formulas in a straightforward way, along the same lines as before. Note that universal quantification must be taken over the appropriate domain, so we have:

   $(M, s, v) \models \forall x^o \varphi$ iff $(M, s, v[x^o/d]) \models \varphi$ for all $d \in D_s$.
4. We have assumed that there is only one probability measure μ on the set of states. We may want to allow uncertainty about the probability functions. We can achieve this by associating with each state a (possibly different) probability function on the set of states (cf. [FH88, Hal89]). Thus a structure would now consist of a tuple $(D, S, \pi, \{\mu^s : s \in S\})$; in order to evaluate the value of the (field) term $w(\varphi)$ in a state s, we use the probability function μ^s.

4 Probabilities on the domain and on possible worlds

In the previous sections we have presented structures to capture two different modes of probabilistic reasoning.[2] We do not want to say that one mode is more "right" than another; they both have their place. Clearly there might be situations where we want to do both modes of reasoning simultaneously. We consider three examples here.

Example 4.1: Consider the statement "the probability that Tweety files is greater than the probability that a random bird flies." This can be captured by the formula

$$w(Flies(Tweety)) > w_x(Flies(x)).$$

Example 4.2: For a more complicated example, consider two statements like "The probability that a random bird flies is greater than .99" and "The probability that a random bird flies is greater than .9." An agent might consider the first statement rather unlikely to be true, and so take it to hold with probability less than .2, while he might consider the second statement exceeding likely to be true, and so take it to hold with probability greater than .95. We can capture this by combining the syntax of the previous two sections to get:

$$w(w_x(Flies(x) \mid Bird(x)) > .99) < .2) \wedge$$
$$w(w_x(Flies(x) \mid Bird(x)) > .90) > .95),$$

where a conditional statement such as $w_x(Flies(x) \mid Bird(x)) > r$ is an abbreviation for $w_x(Flies(x) \wedge Bird(x)) > r w_x(Bird(x))$.[3]

Example 4.3: A final example is given by *Miller's principle* [Sky80], which can be viewed as a way of connecting degrees of belief with probabilities on the domain. Perhaps the most obvious connection we can expect to hold between an agent's degree of belief in $\varphi(\mathrm{a})$, for a particular constant a, and the probability that $\varphi(x)$ holds for a random individual x is equality, as characterized by the following equation:

$$w(\varphi(\mathrm{a})) = w_x(\varphi(x)). \quad (*)$$

Miller's principle says something a little different; it says that for any real number r_0, the conditional probability of $\varphi(\mathrm{a})$, given that the probability that a random x satisfies φ is r_0, is itself r_0. Assuming that the real variable r does not appear free in φ, we can express (this instance of) Miller's principle in our notation as

$$\forall r[w(\varphi(\mathrm{a}) \mid (w_x(\varphi(x)) = r)) = r].$$

We examine the connection between Miller's principle and $(*)$ after we define our formal semantics.

Given a set Φ of function and predicate symbols let $\mathcal{L}_3(\Phi)$ be the language that results by allowing probability terms both of the form $w_{\vec{x}}(\varphi)$, where $\vec{x}$ is a vector of distinct object variables, and of the form $w(\varphi)$; we take $\mathcal{L}_3^=(\Phi)$ to be the extension of $\mathcal{L}_3(\Phi)$ that includes equality between object terms. To give semantics to formulas in $\mathcal{L}_3(\Phi)$ (resp. $\mathcal{L}_3^=(\Phi)$), we will clearly need probability functions over both the set of states and over the domain.

[2] We remark that there is a sense in which we can translate back and forth between domain-based probability and possible-world-based probability. For example, there is an effective translation that maps a formula φ in $\mathcal{L}_1^=$ to a formula φ' in language $\mathcal{L}_2^=$, and a mapping from type 1 structures M to type 2 structure M' such that $M \models \varphi$ iff $M' \models \varphi'$. Similar mappings exist in the other direction. (Details can be found in [AH89].) However, we would still argue that $\mathcal{L}_1$ is not the right language for reasoning about probability over possible worlds, while $\mathcal{L}_2$ is not the right language for reasoning about probability over the domain.

[3] More typically, the conditional probability of A given B is taken to be the probability of $A \cap B$ divided by the probability of B. We have cleared the denominator here to avoid having to deal with the difficulty of dividing by 0 should the probability of B be 0.

Let a *type. S probability structure* be a tuple of the form $(D, S, \pi, \mu_D, \mu_S)$, where *I),* 5, and π are as for type 2 probability structures, μ_D is a discrete probability function on *I)* and μ_S is a discrete probability function on *S.* Intuitively, type 3 structures are obtained by combining type 1 and type 2 structures.

Given a type 3 probability structure M, a state s, and valuation v, we can give semantics to terms and formulas along much the same lines as in type 1 and type 2 structures. For example, we have:

- $[w_{(x_1,\ldots,x_n)}(\varphi)]_{(M,s,v)} = \mu_D^n(\{(d_1,\ldots,d_n) : (M,s,v[x_1/d_1,\ldots,x_n/d_n]) \models \varphi\})$.
- $[w(\varphi)]_{(M,s,v)} = \mu_S(\{s' \in S : (M,s',v) \models \varphi\})$.

As it stands, in a given type 3 probability structure, we have one fixed probability function on the domain. This means that the truth of a formula such as $w_x(\varphi(x)) = r$ is independent of the state; it is either true in all states or false in all states. Thus (using $\models_3$ to denote validity in type 3 structures) we immediately get:

Lemma 4.4:

$\models_3 \forall r[(w(w_x(\varphi(x)) = r) = 1) \vee (w(w_x(\varphi(x)) = r) = 0)]$.

This means that type 3 structures as we have defined them are not expressive enough to capture the intuition behind Example 4.2, since the conditional probability will be the same in all states. The only way for the agent to believe that the statement "The probability that a random bird flies is greater than .99" holds with probability less than .2 is for the statement to be false at all worlds (and thus hold with probability 0). Similarly, the only way for the agent to believe that the statement "The probability that a random bird flies is greater than .90" holds with probability greater than .95 is for it to be true at all possible worlds.

We can easily extend type 3 structures to deal with this problem. We simply allow the probability function on the domain to be a function on the state; thus at each state *s* we would have a (possibly different) probability function μ_D^s on the domain. When computing the value of a field term such as $w_x(\varphi(x))$ at state .9, we use the function μ_D^s. With this change, a statement such as "The probabilty that a random bird flies is greater than .99" can be true at some worlds and false at others, thus allowing us to better capture our intuitions here. Other extensions of type 3 structures, along the lines discussed for type 1 and type 2 structures, are possible as well.

In type 3 structures as we have defined them, there is a close connection between Miller's principle and (*). In fact, as the following theorem shows, they are equivalent.

Proposition 4.5 : $\models_3 [w(\varphi(\mathbf{a})) = w_x(\varphi(x))] \equiv \forall r[w(\varphi(\mathbf{a}) \mid (w_x(\varphi(x)) = r)) = r]$.

We remark that this result depends crucially on the fact that the probability on the domain is the same at ever}' state. If we drop this assumption, then neither direction of the implication necessarily holds.

The idea of there being two types of probability has arisen in the literature before. Skyrms [Sky80] talks about first and second-order probabilities, where first-order probabilities represent propensities or frequency essentially statistical information while second order probabilities represent degrees of belief. These are called first- and second-order probabilities since typically one has a degree of belief about statistical information (this is the case in our second example above). Although $\mathcal{L}_3(\Phi)$ allows arbitrary alternation of the two types of probability, the semantics does support the intuition that these really are two fundamentally *different* types of probability.

5 On obtaining complete axiomatizations

In order to guide (and perhaps help us automate) our reasoning about probabilities, it would be nice to have a complete deductive system. Unfortunately, results of [AII89] show that in general we will not be able to obtain such a system. We briefly review the relevant results here, and then show that we can obtain complete axiomatizations for important special cases.

5.1 Decidability and undecidability results

All the results in this subsection are taken from [A 1189], The first result is positive:

Theorem 5.1: *If Φ consists only of unary predicates, then the validity problem Jor $\mathcal{L}_1(\Phi)$ with respect to type 1 probability structures is decidable.*

The restrictions made in the previous result (to a language with only unary predicates, without equality between object terms) are both necessary. Once we allow equality in the language, the validity problem is no longer recursively enumerable (r.e.), even if Φ is empty. And a binary predicate in Φ is enough to guarantee that the validity problem is not r.e., even without equality between object terms.

Theorem 5.2:

I. For all Φ, the validity problem Jor $\mathcal{L}_1^=(\Phi)$ with respect to type 1 structures is not r.e.

2. // Φ contains at least one predicate of artty greater than or equal to two, then the validity problem for $\mathcal{L}_1(\Phi)$ with respect to type 1 probability structures is not r.e.

Once we move to $\mathcal{L}_2$, the situation is even worse. Even with only one unary predicates in Φ, the validity problem for $\mathcal{L}_2(\Phi)$ is not r.e. If we have equality, then the validity problem is not r.e. as long as Φ has at least one constant symbol. (Note that $\varphi \Rightarrow (w(\varphi) = 1)$ is valid if φ contains no nonlogical symbols that is, φ does not contain any function or predicate symbols, other than equality so we cannot make any nontrivial probability statements if Φ is empty.)

Theorem 5.3:

I. // Φ contains at least one predicate of arity greater than or equal to one, then the validity problem for $\mathcal{L}_2(\Phi)$ with respect to type S probability structures is not r.e.

2. If Φ is nonempty, then the validity problem for $\mathcal{L}_2^=(\Phi)$ with respect to type 2 probability structures is not r.e.

These results paint a rather discouraging picture as far as complete ax Somatizations go. If a logic is to have a complete recursive axiornatization, then the set of valid formulas must be r.e. (we can enumerate them by just carrying out all possible proofs). Thus, for all the cases cited in the previous theorems for which the validity problem is not r.e., there can be no complete axiornatization.[4]

There is some good news in this bleak picture. In many applications it suffices to consider structures of size bounded by some N (or of size exactly N). In this case, we get decidability.

Theorem 5.4: *// we restrict to finite structures of size at most N then, for all* Φ, *the validity problem for* $\mathcal{L}_1^=(\Phi)$ *(resp.,* $\mathcal{L}_2^=(\Phi)$, $\mathcal{L}_3^=(\Phi)$*) with respect to type 1 (resp., type 2, type 3) probability structures is decidable.*

A fortiori, the same result holds if equality is not in the language. We also get decidability if we restrict to structures of size exactly N.

The restriction to bounded structures is necessary though.

Theorem 5.5: *For all* Φ *(resp., for all nonempty* Φ, *for all* Φ*) the validity problem for* $\mathcal{L}_1^=(\Phi)$ *(resp.,* $\mathcal{L}_2^=(\Phi)$, $\mathcal{L}_3^=(\Phi)$*) with respect to type 1 (resp., type 2, type S) probability structures of finite size is not r.e.*

5.2 An axiom system for probability on the domain

Although the previous results tell us that we cannot in general get a complete axiornatization for reasoning about probability, it is still useful to obtain a collection of sound axioms that lets us carry out much of our rea soning.

In order to carry out our reasoning, we will clearly need axioms for doing first-order reasoning. In order to reason about probabilities, which we take to be real numbers, we need the theory of real closed fields. An *ordered field* is a field with a linear ordering <. A *real closed field* is an ordered field where every' positive element has a square root and every polynomial of odd degree has a root. Tarski showed [Tar51, Sho67] that the theory of real closed fields coincides with the theory of the reals (for the first-order language with equality and nonlogical symbols +, x, >,0, 1). That is, a first-order formula involving these symbols is true of the reals if and only if it is true in every real closed field. lie also showed that the theory of real closed fields is decidablc and has an elegant complete axiornatization. We incorporate this into our axiornatization too, since the language of real closed fields is a sublanguage of $\mathcal{L}_1(\Phi)$.

Our axiom system for reasoning about probabilities on the domain, *AX*, includes the axioms of first order logic, the axioms of real closed fields, and axioms for reasoning about probabilities, similar to those of [Bac88b, FI1M88]. The axioms for reasoning about probability are:

[4] We remark that in [AH89], the exact degree of undecidability of the validity problem for all these logics is completely characterized. It turns out to be wildly undecidable, much harder than the validity problem for the first-order theory of arithmetic. We refer the reader to [ATI89] for details.

Reasoning about probabilities over the domain:

PDI $\forall x_1 \ldots \forall x_n \varphi \Rightarrow w_{\langle x_1,\ldots,x_n\rangle}(\varphi) = 1$, where $(x_1, \ldots, x_n)$ is a sequence of distinct object variables

PD2 $w_{\vec{x}}(\varphi) \geq 0$

PD3 $w_{\vec{x}}(\varphi \wedge \psi) + w_{\vec{x}}(\varphi \wedge \neg\psi) = w_{\vec{x}}(\varphi)$

PD4 $w_{\vec{x}}(\varphi) = w_{\vec{x}[x_i/z]}(\varphi[x_i/z])$, **where** z is an object variable which does not appear in $\vec{x}$ or

PD5 $w_{\vec{x},\vec{y}}(\varphi \wedge \psi) = w_{\vec{x}}(\varphi) \times w_{\vec{y}}(\psi)$, if none of the free variables o1φ re contained in y, none of the free variables of ψ are contained in x, and x and $\vec{y}$ are disjoint

RPD1 From $\varphi \equiv \psi$ **infer** $w_{\vec{x}}(\varphi) = w_{\vec{x}}(\psi)$

Note that PD4 allows us to rename bound variables, while PD5 lets us do reasoning based on the indepen dence of the random variables. *AX* is a straightforward extension of the axiom system used in [F1IM88] for rea soning about the propositional case. Not surprisingly, it is also quite similar to the collection of axioms given in [Bac88b]. Bacchus does not use the axioms for real closed fields, but instead he uses the axioms for ordered fields, since he allows his probability functions to take values in arbitrary ordered fields. His axioms for reasoning about probabilities are essentially the same as ours (indeed, axioms PDI, PD2, and PD4 are also used by Bacchus, while PD5 is a weaker version of one of his axioms).

It is easv to check that these axioms are *sound* with respect to type I probability structures: if M is a type 1 probability structure, then $M \models \varphi$ for every axiom φ. By the results of Subsection 5.1, *AX* (or any other axiom system!) cannot hope to be complete for $\mathcal{L}_1(\Phi)$ once has a predicate of arity at least two, nor can it be complete for $\mathcal{L}_1^=$. However, if we restrict Φ to consist only of unary predicates and do not have equality between object terms in the language, then it is complete.

Theorem 5.6: *If* Φ *consists only of unary predicates, then AX\ is a sound and complete axiornatization for the language* $\mathcal{L}_1(\Phi)$ *with respect to type 1 probability structures.*

The previous result shows that *AX* is rich enough to let us carry out a great deal of probabilistic reasoning. The next result reinforces this impression.

Let AX_1^N be AX_1 together with the following axiom, which says that the domain has size at most TV:

FIN$_N$ $\exists x_1 \ldots x_N \forall y (y = x_1 \vee \ldots \vee y = x_N)$

Theorem 5.7: *AX^N_1 is a sound and complete axiomatization for $L_1^=(Q>)$ with respect to type 1 probability structures of size at most N, for any set Φ.*

We can of course modify axiom FIN$_N$v to say that the domain has exactly TV elements, and get a complete axiornatization with respect to structures of size exactly N. We can also get sound axiom systems AX_2 and AX_3 for reasoning about type 2 and type 3 structures respectively. Again, they are not complete, but they are complete with respect to structures of size at most N once we add FJN$_N$- We leave details to the full paper.

6 Conclusions

We have provided natural semantics to capture two different kinds of probabilistic reasoning: in one, the probability is on the domain, and in the other, the probability is on a set of possible worlds. We also showed how these two modes of reasoning could be combined in one framework. We then considered the problem of providing sound and complete axioms to characterize first-order reasoning about probability. While complexity results of [AH89] show that in general there cannot be a complete axiomatization, we did provide sound axiom systems that we showed were rich enough to enable us to carry out a great deal of interesting probabilistic reasoning. In particular, together with an axiom guaranteeing finitcness, our axiom systems were shown to be complete for domains of bounded size.

Our results form an interesting contrast to those of Bacchus [Bac88b]. Bacchus gives a complete axiomatization for his language (which, as we remarked above, is essentially the same as our language $L_1(\Phi)$ for reasoning about probabilities on the domain), thus showing that the validity problem for his language is r.e. The reason for this difference is that Bacchus allows nonstan dard probability functions, which are only required to be finitely additive and can take values in arbitrary ordered fields. In [A1189] it is shown that all the undecidability results mentioned above can be proved even if we only require the probability function to be finitely additive, and restrict probabilities to taking only rational values. This shows that the key reason that Bacchus is able to obtain a complete axiomatization is that he allows probabilities to take values in arbitrar}' ordered fields.[5]

The situation here is somewhat analogous to that of axiomatizing arithmetic. Godel's famous incompleteness result shows that the first-order theory of arithmetic (for the language with equality and nonlogical symbols +,x,0, 1, where the domain is the natural numbers) does not have a complete axiomatization. The axioms of Pcano Arithmetic are sound for arithmetic, but not complete. They are complete with respect to a larger class of domains (including so-called *nonstandard models).* Our results show that reasoning about probabilities is even *harder* than reasoning about arithmetic (since the validity problem for arithmetic is easier than IIj), and so cannot have a complete axiomatization. However, Bacchus' axioms are complete with respect to a larger class of structures, where probabilities can assume nonstandard values. And just as the axioms of Peano Arithmetic are sufficiently rich to let us carry out a great deal of interesting arithmetic reasoning, so the axioms that we have provided (or the axioms of [Bac88b]) are sufficiently rich to enable us to carry out a great deal of interesting probabilistic reasoning.

[5] Bacchus claims [Bac88b] that it is impossible to have a complete proof theory for countably additive probability functions. Although, as our results show, his claim is essentially correct (at least, as long as the language contains one binary predicate symbol or equality), the reason that he gives for this claim, namely, that the corresponding logic is not compact, is not correct. For example, even if $\Phi = \{P\}$, where P is a unary predicate, the logic is not compact. (Consider the set $\{w_x(P(x)) \neq 0,\ w_x(P(x)) < 1/2,\ w_x(P(x)) < 1/3,\ w_x(P(x)) \leq 1/4, \ldots\}$. Any finite subset of these formulas is satisfiable, but the full set is not.) However, by Theorem 5.6, the logic in this case has a complete axiomatization.

Acknowledgements

Discussions with Fahiem Bacchus provided the initial impetus for this work. 1 would like to thank Martin Abadi, Fahiem Bacchus, Hon Fagin, Hector Levesque, Joe Nunes, and Moshe Vardi for their helpful comments on earb'er drafts.

References

[AH89] M. Abadi and J.Y. Halpern, Decidability and expressiveness for first-order logics of probability. To appear as IBM Technical Report, 1989.

[Bac88a] F. Bacchus, On probability distributions over possible worlds, *Proceedings oj the Fourth Workshop on Uncertainty in Artificial Intelligence,* 1988, pp. 15 21.

[Bac88b] F. Bacchus, *Representing and reasoning with probabilistic knowledge,* Ph.D. thesis, University of Alberta, 1988. Also issued as Waterloo University Technical Report CS-88 31.

[Car50] R. Carnap, *Logical Foundations of Probability,* University of Chicago Press, 1950.

[FH88] R. Fagin and J. Y. Halpern, Reasoning about knowledge and probability: preliminary report, *Proceedings of the Second Conference on Theoretical Aspects of Reasoning about Knowledge* (M. Y. Vardi, ed.), Morgan Kaufmann, 1988, pp. 277 293.

[FH89] R. Fagin and ,1. Y. Halpern, Uncertainty, belief, and probability, *Eleventh International Joint Conference on Artificial Intelligence (IJCAL89),* 1989.

[FHM88] R. Fagin, J. Y. Halpern, and N. Megiddo, A logic for reasoning about probabilities, *Proc. 3rd JEEE Symp. on Logic in Computer Science,* 1988, pp. 277 291.

[Hal89] J. Y. Halpern, The relationship between knowledge, belief, and certainty, IBM Research Report RJ 6765, 1989.

[Kyb88] 11. E. Kyburg, Jr., Higher order probabilities and intervals, *International Journal of Approximate Reasoning 2,* 1988, pp. 195 209.

[Nil86] N. Nilsson, Probabilistic logic, *Artificial Intelligence* 28, 1986, pp. 71 87.

[Sho67] J. R. Shoenfield, *Mathematical Logic,* Addison-Wesley, 1967.

[Sky80] B. Skyrms, *Causal Necessity,* Yale University Press, 1980.

[Tar5l] A. Tarski, *A decision method for elementary algebra and geometry*^ Univ. of California Press, 2nd edition, 1951.